Log Analytics System

The HTTP log analyzer provides a fast and accurate data analysis for public web pages. Each day after midnight, a daily log file of all interactions for the previous 24 hours is pushed from an external source to a spooling directory. The directory structure for the log file will be '/YYYY/MM/DD'. The log records are in ASCII csv format. Only the interaction records for the previous 90 days will be stored for queries, older records get moved to archive. You can download sample data from here: ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz

In this project, we'll learn how we can use Apache Flume, Apache HDFS, Apache Oozie, and Apache PIG to understand the pattern of Web access to a Web server hosted by NASA. Following tasks need to be accomplished in order:

MileStone1: Analyze Log data to understand the contents and structure. Here is the basic log entry structure:

Remote_ip remote_logname username date_time uri status bytes Referer User-Agent(or browser)

E.g., 127.0.0.1 - - [10/Apr/2007:10:39:11 +0300] "GET / HTTP/1.1" 500 606 "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3) Gecko/20061201 Firefox/2.0.0.3 (Ubuntu-feisty)"

You can find detailed information about apache common log format and combined log format at http://stackoverflow.com/questions/9234699/understanding-apache-access-log .

MileStone2: Define flume configuration to gather streaming log data from spooling directory into HDFS. Verify the correctness.

MileStone3: Perform the following Transformation steps on HDFS data:

a) Extract the Search Engine name from Referrer and store it as a separate field

b) Extract the keywords searched and store them as separate field

c) The date must be transformed from database custom format to ISO format.

d) Transform the data into AVRO format

MileStone4: Develop PIG queries to answer the following questions:

a) **Heavy Hitter Analysis:** A list of the top 50 IP addresses by traffic per hour.

b) **Referrer** Analysis: A list of the top 50 external referrers.

c) **Search term Analysis:** The top 50 search terms in referrals from Bing and Google.

d) **Peak Time Analysis**: Build a script to figure the peak time of accesses based on the total number of accesses. Your script should count the total number of accesses every hour and sort them in a descending order. In this script, you do not consider duplicate detection. Your output file should include, a sorted list of the total number of accesses and time periods accordingly.

e) **Error Analysis:** Build a script to figure out which resource have related most errors. From the unique list of your resources, count the number of non-successful accesses (non-200 code) and provide a list of the resources and number of errors sorted in descending order. Your output file should include, a sorted list of the total number of the non-successful access. Your output should show the URLs of resources associated with these values.

f) **Reporting:** Calculate for each date:

   1) Total # of requests for non-image URI's (non-image to filter out icons and the like)

   2) Total # of bytes served for non-image URI's

   3) Top 10 non-image URI's served for that date, sorted by either num_requests or num_bytes depending on the ORDERING parameter.

MileStone5: Export the result of first query to corresponding relational table in MySQL. (You have to create corresponding table in MySQL before export).

MileStone6: Create an oozie job that adds the log data containing last day worth of interactions into HDFS.